

[illegible]

```

RRRRRRRR      DDDDDDDD      HH      HH      000000      MM      MM      EEEEEEEEEEE
RRRRRRRR      DDDDDDDD      HH      HH      000000      MM      MM      EEEEEEEEEEE
RR      RR      DD      DD      HH      HH      00      00      MMMM      MMMM      EE
RR      RR      DD      DD      HH      HH      00      00      MMMM      MMMM      EE
RR      RR      DD      DD      HH      HH      00      00      MM      MM      EE
RR      RR      DD      DD      HH      HH      00      00      MM      MM      EE
RRRRRRRR      DD      DD      HHHHHHHHHH      00      00      MM      MM      EEEEEEEEE
RRRRRRRR      DD      DD      HHHHHHHHHH      00      00      MM      MM      EEEEEEEEE
RR      RR      DD      DD      HH      HH      00      00      MM      MM      EE
RR      RR      DD      DD      HH      HH      00      00      MM      MM      EE
RR      RR      DD      DD      HH      HH      00      00      MM      MM      EE
RR      RR      DD      DD      HH      HH      00      00      MM      MM      EE
RR      RR      DDDDDDDD      HH      HH      000000      MM      MM      EEEEEEEEEEE
RR      RR      DDDDDDDD      HH      HH      000000      MM      MM      EEEEEEEEEEE

```

```

....
....
....
....

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS

```



```
1 0001 0 MODULE RDHOME (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000',
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: MOUNT Utility Structure Levels 1 & 2
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This routine reads the home block, if any, of the volume being mounted.
37 0037 1
38 0038 1 ENVIRONMENT:
39 0039 1
40 0040 1 STARLET operating system, including privileged system services
41 0041 1 and internal exec routines.
42 0042 1
43 0043 1 --
44 0044 1
45 0045 1
46 0046 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 13-Oct-1977 21:29
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1
51 0051 1 V02-004 STJ0002 Steven T. Jeffreys, 29-Aug-1980
52 0052 1 Liberal re-write to facilitate operator assisted mount.
53 0053 1 As a result, the code indentation is screwed up.
54 0054 1
55 0055 1 V02-003 ACG0167 Andrew C. Goldstein, 18-Apr-1980 13:39
56 0056 1 Previous revision history moved to MOUNT.REV
57 0057 1 **
```


RDHOME
V04-000

C 3
16-Sep-1984 01:53:15
14-Sep-1984 12:35:19

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[INIT.SRC]RDHOME.B32;1 Page 2 (1)

```
: 58      0058 1
: 59      0059 1
: 60      0060 1 LIBRARY 'SYSS$LIBRARY:LIB.L32';
: 61      0061 1 REQUIRE 'SRC$:INIDEF.B32';
: 62      0352 1
: 63      0353 1
: 64      0354 1 FORWARD ROUTINE
: 65      0355 1 READ_HOMEBLOCK,
: 66      0356 1 SET_VALID,
: 67      0357 1 CLEAR_VALID;
```

```
: read the home block
: set volume valid bit
: clear volume valid bit
```

EX

Mo
--
LI
IN
IN
IN
ST
ST
CL
DC
RP
PA
CH
CO
DI
UP
HP
VA
CV
JB
SY
SY
RM
SY

```

69 0358 1 GLOBAL ROUTINE READ_HOMEBLOCK (VOLUME_LABEL, FULL_SEARCH) =
70 0359 1
71 0360 1 ++
72 0361 1
73 0362 1 FUNCTIONAL DESCRIPTION:
74 0363 1
75 0364 1 This routine reads the home block, if any, of the volume being mounted.
76 0365 1 Various statuses are returned to be interpreted in the context of the
77 0366 1 operation; errors that are clear losers are signalled.
78 0367 1
79 0368 1
80 0369 1 CALLING SEQUENCE:
81 0370 1 READ_HOMEBLOCK (ARG1, ARG2)
82 0371 1
83 0372 1 INPUT PARAMETERS:
84 0373 1 ARG1: address of volume label string descriptor
85 0374 1 ARG2: 1 if search whole disk for home block
86 0375 1 0 if limited search
87 0376 1
88 0377 1 IMPLICIT INPUTS:
89 0378 1 CHANNEL: channel number assigned to device being mounted
90 0379 1
91 0380 1 OUTPUT PARAMETERS:
92 0381 1 NONE
93 0382 1
94 0383 1 IMPLICIT OUTPUTS:
95 0384 1 HOME_BLOCK: buffer contains home block if found
96 0385 1 HOMEBLOCK_LBN: LBN of home block read
97 0386 1
98 0387 1 ROUTINE VALUE:
99 0388 1 1 if valid and correct home block found
100 0389 1 $$$_NOHOMEBLK if home block not found
101 0390 1 $$$_INCVOLLABEL if home block found but wrong volume name
102 0391 1 $$$_DEVOFFLINE
103 0392 1 $$$_MEDOFL
104 0393 1 $$$_FILESTRUCT
105 0394 1
106 0395 1 SIDE EFFECTS:
107 0396 1 NONE
108 0397 1
109 0398 1 --
110 0399 1
111 0400 2 BEGIN
112 0401 2
113 0402 2 MAP
114 0403 2 VOLUME_LABEL : REF VECTOR; ! volume label string descriptor
115 0404 2
116 0405 2 LOCAL
117 0406 2 STATUS, ! system service status
118 0407 2 IO STATUS : VECTOR[2], ! I/O status block
119 0408 2 DELTA, ! home block search delta
120 0409 2 BLOCKFACT, ! device blocking factor
121 0410 2 LBN; ! current LBN being tried
122 0411 2
123 0412 2 EXTERNAL
124 0413 2 CHANNEL, ! channel number for I/O
125 0414 2 DEVICE_CHAR : BBLOCK, ! disk device characteristics
```



```

126      0415      2      DEVCHAR_DESC      : VECTOR,      ! device characteristics descriptor
127      0416      2      HOME_BLOCK      : BBLOCK,      ! home block buffer
128      0417      2      HOME_BLOCK_LBN,      ! LBN of home block read
129      0418      2      CTL$GL_PHD      : REF BBLOCK ADDRESSING_MODE (ABSOLUTE);
130      0419      2      ! pointer to process header
131      0420      2
132      0421      2      EXTERNAL ROUTINE
133      0422      2      READ_BLOCK,      ! read a disk block
134      0423      2      CHECK_HOME_BLK1,      ! verify structure level 1 home block
135      0424      2      CHECK_HOME_BLK2;      ! verify structure level 2 home block
136      0425      2
137      0426      2
138      0427      2
139      0428      2      ! Issue a pack acknowledge so we can talk to the disk.
140      0429      2      ! Inhibit error logging to avoid saturating the error log
141      0430      2      ! with media/device offline errors. These errors cause
142      0431      2      ! us to return and try an operator assisted mount.
143      0432      2      !
144      0433      2
145      P 0434      2      STATUS = $QIOW (CHAN = .CHANNEL,
146      P 0435      2      FUNC = (IO$PACKACK OR IO$M_INHERLOG),
147      0436      2      IO$B = IO_STATUS[0]);
148      0437      2      IF .STATUS THEN STATUS = -(IO_STATUS[0])<0,16>;
149      0438      2      IF NOT .STATUS
150      0439      2      THEN
151      0440      2      !
152      0441      2      ! Allow ILLIOFUNC errors, as devices that have
153      0442      2      ! no PACKACK function will return this status.
154      0443      2      !
155      0444      2      IF .STATUS NEQ SS$_ILLIOFUNC
156      0445      2      THEN
157      0446      2      ERR_EXIT (.STATUS);
158      0447      2
159      0448      2      ! Set the volume valid bit to indicate presence of a volume.
160      0449      2      !
161      0450      2
162      0451      2      KERNEL_CALL (SET_VALID);
163      0452      2
164      0453      2      ! Re-read the device characteristics. (For some devices, packack causes the
165      0454      2      ! correct characteristics to be determined and recorded in the I/O data base.)
166      0455      2      !
167      0456      2
168      0457      2      $GETCHN (CHAN = .CHANNEL, PRIBUF = DEVCHAR_DESC);
169      0458      2
170      0459      2      ! Compute the home block search delta from the volume geometry in the
171      0460      2      ! device table. This is done according to the following rules, where volume
172      0461      2      ! geometry is expressed in the order sectors, tracks, cylinders:
173      0462      2      !
174      0463      2      n x 1 x 1:      1
175      0464      2      1 x n x 1:      1
176      0465      2      1 x 1 x n:      1
177      0466      2      !
178      0467      2      n x m x 1:      n+1
179      0468      2      n x 1 x m:      n+1
180      0469      2      1 x n x m:      n+1
181      0470      2      !
182      0471      2      s x t x c:      (t+1)*s+1
```



```
183 0472 2 !
184 0473 2
185 0474 3 BLOCKFACT = (.DEVICE_CHAR[DIB$B_SECTORS]
186 0475 3 * .DEVICE_CHAR[DIB$B_TRACKS]
187 0476 3 * .DEVICE_CHAR[DIB$W_CYLINDERS])
188 0477 2 / .DEVICE_CHAR[DIB$L_MAXBLOCK];
189 0478 2
190 0479 2 DELTA = 1;
191 0480 2 IF .DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1
192 0481 2 AND .DEVICE_CHAR[DIB$B_TRACKS] GTR 1
193 0482 2 THEN DELTA = .DELTA + .DEVICE_CHAR[DIB$B_TRACKS];
194 0483 2
195 0484 2 IF .DEVICE_CHAR[DIB$B_SECTORS] GTR 1
196 0485 3 AND (.DEVICE_CHAR[DIB$W_CYLINDERS] GTR 1
197 0486 3 OR .DEVICE_CHAR[DIB$B_TRACKS] GTR 1)
198 0487 2 THEN DELTA = (.DELTA * .DEVICE_CHAR[DIB$B_SECTORS] + .BLOCKFACT) / .BLOCKFACT;
199 0488 2
200 0489 2 IF .DELTA EQL 0
201 0490 2 OR .DELTA GTRU .DEVICE_CHAR[DIB$L_MAXBLOCK] / 10
202 0491 2 THEN DELTA = 1;
203 0492 2
204 0493 2 ! Search for the home block. If the device is being mounted foreign, we
205 0494 2 ! limit the home block search to 10 hits to save time. If the device is
206 0495 2 ! being mounted Files-11 then we try all the way just in case the home block
207 0496 2 ! is in a weird location. Note the potential protection hole: Disks with the
208 0497 2 ! home block far into the disk are not protected from being mounted foreign
209 0498 2 ! by non-privileged users. C'est la vie.
210 0499 2 !
211 0500 2
212 0501 2 LBN = 1;
213 0502 2
214 0503 3 IF (
215 0504 3 DECRU J FROM (IF .FULL_SEARCH THEN -1 ELSE 10) TO 1 DO
216 0505 4 BEGIN
217 0506 4 STATUS = READ_BLOCK (.LBN, HOME_BLOCK);
218 0507 4 IF .STATUS
219 0508 4 THEN
220 0509 5 BEGIN
221 0510 5 IF .HOME_BLOCK[HM2$B_STRUCLEV] EQL 1
222 0511 5 THEN STATUS = CHECK_HOME_BLK1 (HOME_BLOCK, .LBN, .VOLUME_LABEL)
223 0512 5 ELSE IF .HOME_BLOCK[HM2$B_STRUCLEV] EQL 2
224 0513 5 THEN STATUS = CHECK_HOME_BLK2 (HOME_BLOCK, .LBN, .VOLUME_LABEL)
225 0514 5 ELSE STATUS = 0;
226 0515 5 IF .STATUS
227 0516 5 OR .STATUS EQL SSS_INCVOLLABEL
228 0517 5 THEN EXITLOOP 0;
229 0518 5 END
230 0519 4 ELSE
231 0520 5 BEGIN
232 0521 5 IF .STATUS EQL SSS_ILLBLKNUM
233 0522 5 THEN EXITLOOP -1;
234 0523 5 IF .STATUS NEQ SSS_PARITY
235 0524 5 AND .STATUS NEQ SSS_FORMAT
236 0525 5 AND .STATUS NEQ SSS_DATACHECK
237 0526 5 THEN ERR_EXIT (.STATUS);
238 0527 4 END;
239 0528 4
```

```

: 240      0529 4      IF NOT .FULL SEARCH
: 241      0530 4      AND .BBLOCK [CTL$GL PHD[PHD$Q_PRIVMSK], PRV$V_VOLPRO]
: 242      0531 4      THEN RETURN (SS$_NOHOMEBLK);
: 243      0532 4
: 244      0533 4      LBN = .LBN + .DELTA;
: 245      0534 4      END
: 246      0535 4      )
: 247      0536 3      THEN
: 248      0537 3      BEGIN
: 249      0538 3      READ_BLOCK (1, HOME_BLOCK);
: 250      0539 3      RETURN (SS$_NOHOMEBLK);
: 251      0540 2      END;
: 252      0541 2
: 253      0542 2      HOMEBLOCK_LBN = .LBN;
: 254      0543 2
: 255      0544 2      RETURN .STATUS;
: 256      0545 2
: 257      0546 1      END;
! end of routine READ_HOMEBLK
```

```

.TITLE RDHOME
.IDENT \V04-000\

.EXTRN CHANNEL, DEVICE CHAR
.EXTRN DEVCHAR_DESC, HOME_BLOCK
.EXTRN HOMEBLOCK_LBN, CTL$GL PHD
.EXTRN READ_BLOCK, CHECK_HOMEBLK1
.EXTRN CHECK_HOMEBLK2, SYSS$QIOW
.EXTRN SYSS$CMKRNL, SYSS$GETCHN

.PSECT $CODE$,NOWRT,2

.ENTRY READ_HOMEBLOCK, Save R2,R3,R4,R5,R6,R7,R8,- R9
03FC 00000
59 00000000G 00 9E 00002 MOVAB LIB$STOP, R9
58 0000G CF 9E 00009 MOVAB HOME_BLOCK, R8
57 0000G CF 9E 0000E MOVAB DEVICE_CHAR+10, R7
5E 08 02 00013 SUBL2 #8, SP
7E 7C 00016 CLRQ -(SP)
7E 7C 00018 CLRQ -(SP)
7E 7C 0001A CLRQ -(SP)
7E 7C 0001C CLRQ -(SP)
7E AE 9F 0001E PUSHAB IO STATUS
7E 0808 8F 3C 00021 MOVZWL #2056, -(SP)
00000000G 00 CF DD 00026 PUSHL CHANNEL
53 50 DO 0002C CLRL -(SP)
06 53 E9 00036 CALLS #12, SYSS$QIOW
53 6E 3C 00039 MOVL R0, STATUS
0E 53 E8 0003C BLBC STATUS, 1$
000000F4 8F 53 D1 0003F 1$: BLBS STATUS, 2$
05 13 00046 CMPL STATUS, #244
53 DD 00048 BEQL 2$
69 01 FB 0004A PUSHL STATUS
7E D4 0004D 2$: CALLS #1, LIB$STOP
5E DD 0004F CLRL -(SP)
PUSHL SP
```


00000000G	9F	0000V	CF	9F	00051	PUSHAB	SET_VALID	:	
			03	FB	00055	CALLS	#3, @#SYSS\$CMKRNL	:	
			7E	7C	0005C	CLRQ	-(SP)	:	0457
		0000G	CF	9F	0005E	PUSHAB	DEVCHAR_DESC	:	
			7E	D4	00062	CLRL	-(SP)	:	
		0000G	CF	DD	00064	PUSHL	CHANNEL	:	
			05	FB	00068	CALLS	#5, SYSS\$GETCHN	:	
		FE	A7	9A	0006F	MOVZBL	DEVICE_CHAR+8, R0	:	0474
		FF	A7	9A	00073	MOVZBL	DEVICE_CHAR+9, R1	:	0475
52			51	C5	00077	MULL3	R1, R0, R2	:	
			67	3C	0007B	MOVZWL	DEVICE_CHAR+10, R4	:	0476
			54	C4	0007E	MULL2	R4, R2	:	
54		66	A7	C7	00081	DIVL3	DEVICE_CHAR+112, R2, BLOCKFACT	:	0477
			01	D0	00086	MOVL	#1, DELTA	:	0479
			55	D4	00089	CLRL	R5	:	0480
			67	B1	0008B	CMPW	DEVICE_CHAR+10, #1	:	
			0A	1B	0008E	BLEQU	3\$	:	
			55	D6	00090	INCL	R5	:	
		01	51	91	00092	CMPB	R1, #1	:	0481
			03	1B	00095	BLEQU	3\$	:	
		52	51	C0	00097	ADDL2	R1, DELTA	:	0482
		01	50	91	0009A	CMPB	R0, #1	:	0484
			12	1B	0009D	BLEQU	5\$	:	
		05	55	E8	0009F	BLBS	R5, 4\$	:	0485
		01	51	91	000A2	CMPB	R1, #1	:	0486
			0A	1B	000A5	BLEQU	5\$	:	
		50	52	C4	000A7	MULL2	DELTA, R0	:	0487
		50	54	C0	000AA	ADDL2	BLOCKFACT, R0	:	
52		50	54	C7	000AD	DIVL3	BLOCKFACT, R0, DELTA	:	
			52	D5	000B1	TSTL	DELTA	:	0489
			0A	13	000B3	BEQL	6\$	:	
50	66	A7	0A	C7	000B5	DIVL3	#10, DEVICE_CHAR+112, R0	:	0490
		50	52	D1	000BA	CMPL	DELTA, R0	:	
			03	1B	000BD	BLEQU	7\$	:	
		52	01	D0	000BF	MOVL	#1, DELTA	:	0491
		54	01	D0	000C2	MOVL	#1, LBN	:	0501
		05	AC	E9	000C5	BLBC	FULL_SEARCH, 8\$	:	0504
		55	01	CE	000C9	MNEGL	#1, R5	:	
			03	11	000CC	BRB	9\$	:	
		55	0A	D0	000CE	MOVL	#10, R5	:	
		56	AC	D2	000D1	MCOML	FULL_SEARCH, R6	:	0529
			0086	31	000D5	BRW	18\$	:	0533
		0110	8F	BB	000D8	PUSHR	#^M<R4,R8>	:	0506
			02	FB	000DC	CALLS	#2, READ_BLOCK	:	
0000G	CF		50	D0	000E1	MOVL	R0, STATUS	:	
	53		53	E9	000E4	BLBC	STATUS, 15\$	:	0507
	3B					CMPB	HOME_BLOCK+13, #1	:	0510
	01	0D	A8	91	000E7	BNEQ	11\$	:	
			0E	12	000EB	PUSHL	VOLUME_LABEL	:	0511
		04	AC	DD	000ED	PUSHL	LBN	:	
			54	DD	000F0	PUSHL	R8	:	
			58	DD	000F2	PUSHL	R8	:	
0000G	CF		03	FB	000F4	CALLS	#3, CHECK_HOME_BLK1	:	
			12	11	000F9	BRB	12\$	:	
	02	0D	A8	91	000FB	CMPB	HOME_BLOCK+13, #2	:	0512
			11	12	000FF	BNEQ	13\$	:	
		04	AC	DD	00101	PUSHL	VOLUME_LABEL	:	0513
			54	DD	00104	PUSHL	LBN	:	

Address	Hex	Label	Op	Op2	Op3	Op4	Op5	Op6	Op7	Op8	Op9	Op10	Op11	Op12	Op13	Op14	Op15	Op16	Op17	Op18	Op19	Op20	Op21	Op22	Op23	Op24	Op25	Op26	Op27	Op28	Op29	Op30	Op31	Op32	Op33	Op34	Op35	Op36	Op37	Op38	Op39	Op40	Op41	Op42	Op43	Op44	Op45	Op46	Op47	Op48	Op49	Op50	Op51	Op52	Op53	Op54	Op55	Op56	Op57	Op58	Op59	Op60	Op61	Op62	Op63	Op64	Op65	Op66	Op67	Op68	Op69	Op70	Op71	Op72	Op73	Op74	Op75	Op76	Op77	Op78	Op79	Op80	Op81	Op82	Op83	Op84	Op85	Op86	Op87	Op88	Op89	Op90	Op91	Op92	Op93	Op94	Op95	Op96	Op97	Op98	Op99	Op100	Op101	Op102	Op103	Op104	Op105	Op106	Op107	Op108	Op109	Op110	Op111	Op112	Op113	Op114	Op115	Op116	Op117	Op118	Op119	Op120	Op121	Op122	Op123	Op124	Op125	Op126	Op127	Op128	Op129	Op130	Op131	Op132	Op133	Op134	Op135	Op136	Op137	Op138	Op139	Op140	Op141	Op142	Op143	Op144	Op145	Op146	Op147	Op148	Op149	Op150	Op151	Op152	Op153	Op154	Op155	Op156	Op157	Op158	Op159	Op160	Op161	Op162	Op163	Op164	Op165	Op166	Op167	Op168	Op169	Op170	Op171	Op172	Op173	Op174	Op175	Op176	Op177	Op178	Op179	Op180	Op181	Op182	Op183	Op184	Op185	Op186	Op187	Op188	Op189	Op190	Op191	Op192	Op193	Op194	Op195	Op196	Op197	Op198	Op199	Op200	Op201	Op202	Op203	Op204	Op205	Op206	Op207	Op208	Op209	Op210	Op211	Op212	Op213	Op214	Op215	Op216	Op217	Op218	Op219	Op220	Op221	Op222	Op223	Op224	Op225	Op226	Op227	Op228	Op229	Op230	Op231	Op232	Op233	Op234	Op235	Op236	Op237	Op238	Op239	Op240	Op241	Op242	Op243	Op244	Op245	Op246	Op247	Op248	Op249	Op250	Op251	Op252	Op253	Op254	Op255	Op256	Op257	Op258	Op259	Op260	Op261	Op262	Op263	Op264	Op265	Op266	Op267	Op268	Op269	Op270	Op271	Op272	Op273	Op274	Op275	Op276	Op277	Op278	Op279	Op280	Op281	Op282	Op283	Op284	Op285	Op286	Op287	Op288	Op289	Op290	Op291	Op292	Op293	Op294	Op295	Op296	Op297	Op298	Op299	Op300	Op301	Op302	Op303	Op304	Op305	Op306	Op307	Op308	Op309	Op310	Op311	Op312	Op313	Op314	Op315	Op316	Op317	Op318	Op319	Op320	Op321	Op322	Op323	Op324	Op325	Op326	Op327	Op328	Op329	Op330	Op331	Op332	Op333	Op334	Op335	Op336	Op337	Op338	Op339	Op340	Op341	Op342	Op343	Op344	Op345	Op346	Op347	Op348	Op349	Op350	Op351	Op352	Op353	Op354	Op355	Op356	Op357	Op358	Op359	Op360	Op361	Op362	Op363	Op364	Op365	Op366	Op367	Op368	Op369	Op370	Op371	Op372	Op373	Op374	Op375	Op376	Op377	Op378	Op379	Op380	Op381	Op382	Op383	Op384	Op385	Op386	Op387	Op388	Op389	Op390	Op391	Op392	Op393	Op394	Op395	Op396	Op397	Op398	Op399	Op400	Op401	Op402	Op403	Op404	Op405	Op406	Op407	Op408	Op409	Op410	Op411	Op412	Op413	Op414	Op415	Op416	Op417	
---------	-----	-------	----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	--

; Routine Size: 381 bytes, Routine Base: \$CODES + 0000


```
259 0547 1 GLOBAL ROUTINE SET_VALID =
260 0548 1
261 0549 1 ++
262 0550 1
263 0551 1 FUNCTIONAL DESCRIPTION:
264 0552 1
265 0553 1 This routine sets the volume valid bit in the UCB, causing
266 0554 1 PACKACK to be reissued in case of a power failure or volume
267 0555 1 status transition.
268 0556 1
269 0557 1
270 0558 1 CALLING SEQUENCE:
271 0559 1 SET_VALID ()
272 0560 1
273 0561 1 INPUT PARAMETERS:
274 0562 1 NONE
275 0563 1
276 0564 1 IMPLICIT INPUTS:
277 0565 1 CHANNEL: channel number assigned to device
278 0566 1
279 0567 1 OUTPUT PARAMETERS:
280 0568 1 NONE
281 0569 1
282 0570 1 IMPLICIT OUTPUTS:
283 0571 1 NONE
284 0572 1
285 0573 1 ROUTINE VALUE:
286 0574 1 1
287 0575 1
288 0576 1 SIDE EFFECTS:
289 0577 1 valid bit set in UCB
290 0578 1
291 0579 1 --
292 0580 1
293 0581 2 BEGIN
294 0582 2
295 0583 2 LOCAL
296 0584 2 UCB : REF BBLOCK; ! pointer to UCB
297 0585 2
298 0586 2 EXTERNAL
299 0587 2 CHANNEL; ! channel assigned to device
300 0588 2
301 0589 2 EXTERNAL ROUTINE
302 0590 2 GET_CHANNELUCB; ! get UCB of channel
303 0591 2
304 0592 2
305 0593 2 ! Get the UCB address from the channel and set the bit.
306 0594 2 !
307 0595 2
308 0596 2 UCB = GET_CHANNELUCB (.CHANNEL);
309 0597 2 UCB[UCB$V_VALID] = 1;
310 0598 2
311 0599 2 RETURN 1;
312 0600 2
313 0601 1 END; ! end of routine SET_VALID
```

; Routine Size: 19 bytes, Routine Base: \$CODE\$ + 017D


```

315 0602 1 GLOBAL ROUTINE CLEAR_VALID =
316 0603 1
317 0604 1 ++
318 0605 1
319 0606 1 FUNCTIONAL DESCRIPTION:
320 0607 1
321 0608 1 This routine clears the volume valid bit in the UCB.
322 0609 1
323 0610 1
324 0611 1 CALLING SEQUENCE:
325 0612 1 CLEAR_VALID ()
326 0613 1
327 0614 1 INPUT PARAMETERS:
328 0615 1 NONE
329 0616 1
330 0617 1 IMPLICIT INPUTS:
331 0618 1 CHANNEL: channel number assigned to device
332 0619 1
333 0620 1 OUTPUT PARAMETERS:
334 0621 1 NONE
335 0622 1
336 0623 1 IMPLICIT OUTPUTS:
337 0624 1 NONE
338 0625 1
339 0626 1 ROUTINE VALUE:
340 0627 1 1
341 0628 1
342 0629 1 SIDE EFFECTS:
343 0630 1 valid bit clear in UCB
344 0631 1
345 0632 1 --
346 0633 1
347 0634 2 BEGIN
348 0635 2
349 0636 2 LOCAL
350 0637 2 UCB : REF BBLOCK; ! pointer to UCB
351 0638 2
352 0639 2 EXTERNAL
353 0640 2 CHANNEL; ! channel assigned to device
354 0641 2
355 0642 2 EXTERNAL ROUTINE
356 0643 2 GET_CHANNELUCB; ! get UCB of channel
357 0644 2
358 0645 2
359 0646 2 ! Get the UCB address from the channel and clear the bit.
360 0647 2 !
361 0648 2
362 0649 2 UCB = GET_CHANNELUCB (.CHANNEL);
363 0650 2 UCB[UCB$V_VALID] = 0;
364 0651 2
365 0652 2 RETURN 1;
366 0653 2
367 0654 1 END; ! end of routine CLEAR_VALID
```

RDHOME
V04-000

M 3
16-Sep-1984 01:53:15
14-Sep-1984 12:35:19

VAX-11 Bliss-32 V4.0-742
DISK\$VM\$MASTER:[INIT.SRC]RDHOME.B32;1
Page 12
(4)

```

0000G CF 0000G 0000 00000
65 AO 01 DD 00002
50 08 FB 00006
01 08 8A 0000B
01 D0 0000F
04 00012
```

```

.ENTRY CLEAR VALID, Save nothing
PUSHL CHANNEL
CALLS #1, GET CHANNELUCB
BICB2 #8, 101(UCB)
MOVL #1, R0
RET
```

```

: 0602
: 0649
:
: 0650
: 0652
: 0654
```

; Routine Size: 19 bytes, Routine Base: \$CODE\$ + 0190

```

: 368 0655 1
: 369 0656 1 END
: 370 0657 0 ELUDOM
```

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	419	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPI,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	24	0	1000	00:01.9

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:RDHOME/OBJ=OBJ\$:RDHOME MSRC\$:RDHOME/UPDATE=(ENH\$:RDHOME)

```

: Size: 419 code + 0 data bytes
: Run Time: 00:13.1
: Elapsed Time: 00:39.9
: Lines/CPU Min: 3004
: Lexemes/CPU-Min: 25417
: Memory Used: 147 pages
: Compilation Complete
```


0188 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

INPSMB
MAP

INSDEF
SQL

INPSMBMSG
LIS

RSXLBDF
SQL

INSCREATE
LIS

INITIO
LIS

INSTAL

INSTALLS
MAP

INSCMD
CLD

INSPREFIX
REQ

INPSMBCLD
CLD

INPSMB
LIS

INSOLDCMD
CLD

INTUOL
LIS

RDHOME
LIS

INPSMB

INPSMBCLD
LIS

INSCMD
LIS